REMARKS

This Reply is responsive to the Office Action[1] of September 5, 2007. No claims are amended, added or canceled. Claims 1, 5, 9, 14 and 22 are in independent form. Claims 7, 13 and 23 were previously canceled without prejudice or disclaimer. Thus, claims 1-6, 8-12 and 14-22 remain pending.

Claims 1-6 and 8-12 and 14-22 are rejected under 35 U.S.C. § 103(a) as being un-patentable over newly-cited Minear et al. (U.S. Patent No. 5,983,350; hereinafter "Minear") and further in view of newly-cited Mason et al. (U.S. Patent No. 5,668,998 hereinafter "Mason"). The rejection is respectfully traversed because Minear in view of Mason do not disclose or suggest all elements of all pending claims, for at least the following reasons.

Minear discloses a system and method for regulating the flow of messages through a firewall having a network protocol stack which includes an Internet Protocol (IP) layer. An incoming message (i.e., a "datagram") to the IP layer (i.e., "the kernel") is examined to determine if it is encrypted. If not encrypted, the message is passed up the stack to an application level proxy. If the message is encrypted, it is first decrypted at the IP layer and then passed up the stack to the proxy. Decrypting the message is accomplished by executing a process in the IP layer. (Minear, Abstract).

Consider, for example, claim 1:

In a node operative within a network of a plurality of nodes, a method for performing cryptographic-related functions, comprising:

[1] The Office Action may contain a number of statements characterizing the cited references and/or the claims which Applicants may not expressly identify herein. Regardless of whether or not any such statement is identified herein, Applicants do not automatically subscribe to, or acquiesce in, any such statement.

executing an application program in a user space at the node;

receiving an input requiring cryptographic-related processing;

generating a message via the application program based on the input, the message representing one of a predefined set of messages for processing by one of a plurality of cryptographic processing components located in a kernel space within the node, each one of said messages being associated with a respective one of said cryptographic-related functions;

transmitting the message to one of a socket handler and a call handler in kernel space at the node to obtain a transmitted message;

forwarding the transmitted message to a request handler at the node which generates a function call to the cryptographic processing component appropriate for the transmitted message; and

performing the cryptographic-related processing by the cryptographic processing component appropriate for the transmitted message.

The Office Action reads firewall 18 on the recited network "node" by applying Minear, col. 5, lines 34-53 to Applicants' recited: "executing an application program in a user space at the node." (O.A. pg 3) The Office Action also presumably reads "the message coming in" on Applicants' recited "input" requiring crypto processing, such message being included in Minear, col. 6, lines 13-27 which is applied against Applicants' recited: "receiving an input requiring cryptographic-related processing." (O.A. pg 3) The "message coming in" is also referred to in Minear as a "datagram" (*see*, e.g., col. 4, line 47) With the foregoing in mind consider Applicants' message generating step.

The Office Action (pg 3) alleges that the following sections of Minear read on Applicants' message generating step:

> an SADB Master copy 52 is maintained in persistent memory at application layer 48 while a copy 54 of SADB is maintained in volatile memory within the kernel. (Minear, col. 6, lines 7-9)

This section, referring to Fig. 2 in Minear, merely discusses the maintenance of a Security Association Data Base (SADB) master copy in the application layer 48 while a working copy of SADB is maintained in the IP kernel layer.

> In the embodiment shown in FIG. 2, a working copy 54 of the Security Association Database consisting of all currently active SA's is kept resident in memory for ready access by IP layer processing as datagrams are received and transmitted. In addition, a working master copy 52 of the SADB is maintained in a file in nonvolatile memory. During system startup and initialization processing the content of all of the required SA's in master SADB 52 is added to the working copy 54 stored in kernel memory. (Minear, col. 6, lines 33-40; emphasis added)

This section says that SADB 54 in the IP layer is kept ready for processing as input datagrams are received and transmitted. Further, although a working master copy 52 of that SADB is in the application layer, the content of database 54 is initialized at startup/initialization, not "based on the input" as claimed. Moreover, all currently active SA's are processed in the kernel (IP) layer which does not appear to have any bearing on Applicants' message generating step which generates a message "via the application program" in the user layer. If anything, this section refers to processing in the kernel (IP) layer, not in the user (application) layer.

> In one embodiment, a flag which accompanies the message as it is sent from IP layer 44 to proxy 50 indicates whether the incoming message was or was not encrypted. In another embodiment, proxy 50 accesses Security Association Database 54 (the table in the kernel can be queried via an SADB routing socket (PF-SADB)) to determine whether or not a security association exists for a given peer. The SADB socket is much like a routing socket found in the stock BSD 4.4 kernel (protocol family PF-ROUTE) except that PF-SADB sockets are used to maintain the Security Association Database (SADB) instead of the routing table. Because the private keys used for encryption, decryption, and keyed authentication are stored in this table, access must be strictly prohibited and allowed to only administrators and key management daemons. Care must be taken when allowing user-level daemons access to /dev/mem or /dev/kmem as well, since the keys are stored in kernel memory and could be exposed with some creative hacking. (Minear, col. 7, lines 23-40)

This section discusses two embodiments. In one embodiment a flag is sent from the kernel (IP) layer to proxy 50 in user space. This embodiment cannot read on "generating a message via the application program..." as recited in claim 1 because the flag is originated in the IP kernel and not in the application program in user space. In the other embodiment, proxy 50 in user space accesses SADB 54 in kernel space to determine if a security association (SA) exists for a "given peer." A "given peer" is referred to in Minear as relating to **OUTPUT**, not input. For example, "The peer SPI - The SPI value to put on a IPSEC datagram on *output*. The peer IP address - The *destination* IP address to be put into the packet header if IPSEC Tunnel mode is used." (Minear, column 4, lines 16-19, emphasis added) Therefore, the other embodiment in this section teaches that something may be generated by proxy 50 in user space to determine if an SA exists based on an output. Accordingly, the other embodiment of this section also cannot read on Applicants' recited generating step which is based on an input.

Indeed, for at least these reasons, this section of Minear does not read on "generating a message via the application program based on the input, the message representing one of a predefined set of messages for processing by one of a plurality of cryptographic processing components located in a kernel space within the node, each one of said messages being associated with a respective one of said cryptographic-related functions" as recited in claim 1.

In addition, these three sections do not read on Applicants' recited generating step because they also do not disclose or suggest that Minear's expressed message (or any

implied message that may be contained therein) represents one of a predefined set of messages. Applicants' recited message is generated in user space via the application program and is thereafter processed in kernel space by a cryptographic component. This is discussed in Applicants' specification, at least at pgs 11-13. There is no teaching in these sections (or anyplace else in Minear) of a message from a predefined set of inter-protocol-level messages being sent from user space to kernel space in Minear. Therefore, Minear does not disclose or suggest: "generating a message via the application program based on the input, the message representing one of a predefined set of messages for processing by one of a plurality of cryptographic processing components located in a kernel space within the node, each one of said messages being associated with a respective one of said cryptographic-related functions" as recited in claim 1.

Since Applicants' message generating step in claim 1 is not disclosed or suggested by Minear, claim 1 itself is not disclosed or suggested by Minear. Mason, cited to teach a call handler, does not cure this deficiency in Minear. Based on MPEP 2143, therefore, a prima facie case of obviousness has not been established against claim 1 on the basis on Minear in view of Mason. Accordingly, the 35 U.S.C. § 103(a) rejection of claim 1 as being un-patentable over Minear and further in view of Mason should be withdrawn and the claim allowed.

Dependent claims 2-4, dependent from claim 1, are also allowable, at least for reasons based on their respective dependencies from allowable base claim 1.

Turning next to claim 14, it recites, interalia: "generating in said node a predefined message based on the input, the message representing one of a plurality of

predefined messages usable by a cryptographic processing program executed by one of a plurality of cryptographic processing components in kernel space, each one of said messages being associated with a respective one of said cryptographic-related functions." The Office Action cites the same sections in Minear against this generating step as it cited against the generating step of claim 1. Therefore, claim 14 is likewise allowable for reasons given above with respect to claim 1.

Dependent claims 15-21, dependent from claim 14, are also allowable, at least for reasons based on their respective dependencies, directly or indirectly, from an allowable base claim.

Next, consider claim 5. The Office Action cites Minear col. 6, lines 13-27 and col. 11, lines 30 -33 as allegedly reading on "receiving an input representing one of predefined messages." (O.A. pg 6) Applicants respectfully disagree. These sections are reproduced:

> Proxy 50 examines the source and destination addresses and the type of service desired and decides whether authentication of the sender is warranted. If so, proxy 50 initiates an authentication protocol. The protocol may be as simple as requesting a user name and password or it may include a challenge/response authentication process. Proxy 50 also looks to see whether the message coming in was encrypted or not and may factor that into whether a particular type of authentication is needed. In Telnet, for instance, user name/password authentication may be sufficient for an FFE link while the security policy may dictate that a more stringent challenge/response protocol is needed for unencrypted links. In that case, proxy 50 will be a Telnet proxy and it will base its authentication protocol on whether the link was encrypted or not. (Minear, col. 6, lines 13-27, emphasis added)

This section does not disclose or suggest an input representing one of the predefined messages. To begin with, predefined messages are not disclosed or suggested. The only

"message" mentioned is the "message coming in" which is the datagram input. That message is the one being examined in the IP layer (Fig. 2) to determine whether or not it is encrypted, as discussed earlier because it is an unknown upon arrival. Therefore, that datagram is not a predefined message - it is unknown and, therefore, an undefined message.

There is nothing else in this section that can be reasonably interpreted as an input allegedly representing one of predefined messages. The initiation of the authentication protocol by proxy 50 does not imply predefined messages because that initiation requests a user name and password, and it may include a challenge/response, with respect to an examined datagram source or destination address. Therefore, any such request or challenge is inherently not a predefined message because the message is sent to the instant datagram source or destination address that was just examined. That datagram address is a destination address for the "message" and that destination address (thus, the message) cannot be defined until the datagram was received. Clearly, every message has at least a destination and content. A message is predefined when at least its destination and content are both predefined. Message content alone is insufficient to define a message. Without a destination to receive the message, content, by itself, has little or no meaning, is not a complete message and is, therefore, not a predefined message. If the Examiner persists in this rejection, Applicants respectfully request that the Examiner point to precisely the language being interpreted as the alleged predefined messages.

> 10. When a dynamic entry is found by a SPI search, the current datagram's SRC address, which is required to ensure that the return datagrams are properly encrypted, will be recorded in the SA only after the AH checking has passed successfully. (This is because if the address is recorded before AH passes then an

> attacker can cause return packets of an outgoing connection to be transmitted in the clear.) (Minear, col. 11, lines 29-33)

Applicant cannot discern anything in this section which relates to receiving an input representing one of predefined messages. Again, there are no predefined messages disclosed or suggested. If the Examiner persists in this rejection, Applicants respectfully request that the Examiner point to precisely the language being interpreted as the alleged predefined messages. The above two sections, taken together do not disclose or suggest "receiving an input representing one of the predefined messages" as recited in claim 5.

Next, consider Applicants' transmitting step, against which the Office Action cites Minear, col. 11, line 54 to col. 12, line 12 (O.A. pg 6):

> In one embodiment, crypto engine interface 80 is a utility which provides a consistent interface between the software and hardware encryption engines. As shown in FIG. 4, in one such embodiment interface 80 only supports the use of the use of hardware cryptographic engine 84 for IPSEC ESP processing. The significant design issue that interface 80 must deal with is that use of a hardware encryption engine requires that the processing be down [done] in disjoint steps operating in different interrupt contexts as engine 84 completes the various processing steps.
>
> The required information is stored in a request structure that is bound to the IP datagram being processed. The request is of type crypto.sub.-- request.sub.-- t. This structure is quite large and definitely does not contain a minimum state set.
>
> In addition to the definition of the request data structure, this software implementing interface 80 provides two functions which isolate the decision of which cryptographic engine to use. The crypt.sub.-- des.sub.-- encrypt function is for use by the IP output processing to encrypt a datagram. The crypt.sub.-- des.sub.-- decrypt function is for use by the IP input processing to decrypt a datagram. If hardware encryption engine 84 is present and other hardware usage criteria are met the request is enqueued on a hardware processing queue and a return code indicating that the cryptographic processing is in progress is returned. If software engine 82 is used, the return code indicates that the cryptographic processing is complete. (Minear, col. 11, line 54 - col. 12, line 12; emphasis added)

This section is largely a discussion about the datagram, and encrypting and decrypting it. Applicants cannot discern its relevance to the claimed transmitting, to a cryptographic processing module, a function call based on an input representing one of a plurality of predefined messages, the function call requesting cryptographic processing and being executed by the processor in user space. In fact, this section teaches the reverse.

This section teaches that interface 80 (located in kernel space - Minear, Fig. 4) implements two functions for use by IP output and IP input processing in user space. This function flow in Minear is from kernel space to user space. This is the reverse direction of message flow when compared to the direction of Applicants' direction of messages from user space to kernel space. Therefore, this section may relate to cryptographic processing, but it does not disclose or suggest "....instructions which, when executed by a processor in a user space, cause said processor to perform a method comprising:.....transmitting, based on the input, a function call representing a request for cryptographic-related processing to a cryptographic processing module executed by the processor" as recited in claim 5.

In addition, although the term "key" is mentioned in Minear, there is no discussion in Minear of a public key authentication infrastructure (PKAI), per se. PKAI does not appear in Minear. Therefore, all of the limitations recited in claim 5 which include a reference to PKAI are not disclosed or suggested by Minear. In other words, the receiving, transmitting and performing steps, which are limited to being implemented by PKAI, are not disclosed or suggested by Minear for this additional reason.

For any or all of the above reasons, the limitations discussed above in Applicants' claim 5 are not disclosed or suggested by Minear. Mason, cited to teach a call handler,

does not cure this deficiency in Minear. Based on MPEP 2143, therefore, a prima facie case of obviousness has not been established against claim 5 on the basis on Minear in view of Mason. Accordingly, the 35 U.S.C. § 103(a) rejection of claim 5 as being un-patentable over Minear and further in view of Mason should be withdrawn and the claim allowed.

Dependent claims 6 and 8, dependent from claim 5, are also allowable, at least for reasons based on their respective dependencies from allowable base claim 5.

Next, turning to independent claim 9, this claim is allowable for reasons similar to those given above. Claim 9 recites, *interalia*: "a memory configured to store a plurality of cryptographic processing programs in user space on a computer-readable medium, each program being invoked via one of a plurality of predefined messages." The Office Action cites Minear, col. 5, lines 34-53 and col. 6, lines 7-12 against this limitation. (O.A. pg 9) These sections are reproduced:

> In a system such as system 10, application level gateway firewall 18 acts as a buffer between unprotected network 16 and workstations such as workstation 20. Messages coming from unprotected network 16 are reviewed and a determination is made as to whether execution of an authentication and identification protocol is warranted. In contrast to previous systems, system 10 also performs this same determination on IPSEC-encrypted messages. If desired, the same authentication and identification can be made on messages to be transferred from workstation 20 to unprotected network 16. FIG. 2 illustrates one way of authenticating both encrypted and unencrypted messages in a system such as system 10.
> In the system of FIG. 2 a network protocol stack 40 includes a physical layer 42, an Internet protocol (IP) layer 44, a Transport layer 46 and an application layer 48. Such a protocol stack exists, for instance on application level gateway firewall 18 of FIG. 1. An application executing in application layer 48 can communicate to an application executing on another system by preparing a message and transmitting it through one of the existing transport services executing on transport layer 46. (Minear, Col. 5, lines 34-55; emphasis added)

The only messages discussed in this section are those which are equivalent to the datagram. These messages are not the predefined messages that invoke user-space stored programs of claim 9. Predefined messages are not disclosed or suggested in this section at all, much less the specific predefined messages recited in claim 9. If the Examiner persists in this rejection, Applicants respectfully request that the Examiner point to precisely the language being interpreted as the alleged predefined messages.

> In the embodiment shown in FIG. 2, an SADB Master copy 52 is maintained in persistent memory at application layer 48 while a copy 54 of SADB is maintained in volatile memory within the kernel. If the message is supposed to be encrypted, the message is decrypted based on the algorithm and key associated with the particular SA and the message is transferred up through transport layer 46 to proxy 50. (Minear, col. 6, lines 6-13)

Again, the only messages discussed in this section are those equivalent to the datagram, and they are transferred in the direction from IP layer "up through the transport layer" to user space, which is the reverse direction from the direction of Applicants' message. If an incoming datagram is expected or supposed to be encrypted, it is first decrypted before it is forwarded or transferred to proxy 50 in the application layer. This explanation is not at all suggestive of a plurality of predefined messages which invoke programs that have been stored in user space. Accordingly, these sections do not disclose or suggest: "a memory configured to store a plurality of cryptographic processing programs in user space on a computer-readable medium, each program being invoked via one of a plurality of predefined messages" as recited in claim 9. If the Examiner persists in this rejection, Applicants respectfully request that the Examiner point to precisely the language being interpreted as the alleged predefined messages.

In addition, "a processor configured to: generate one of the predefined messages based on the input" as recited in claim 9 is not disclosed or suggested by the applied sections (col. 6, lines 13-27 and col. 11, lines 30-33).

> Proxy 50 examines the source and destination addresses and the type of service desired and decides whether authentication of the sender is warranted. If so, proxy 50 initiates an authentication protocol. The protocol may be as simple as requesting a user name and password or it may include a challenge/response authentication process. Proxy 50 also looks to see whether the message coming in was encrypted or not and may factor that into whether a particular type of authentication is needed. In Telnet, for instance, user name/password authentication may be sufficient for an FFE link while the security policy may dictate that a more stringent challenge/response protocol is needed for unencrypted links. In that case, proxy 50 will be a Telnet proxy and it will base its authentication protocol on whether the link was encrypted or not. (Minear, col. 6, lines 13-27; emphasis added)

This section does not disclose or suggest an input representing one of the predefined messages. To begin with, predefined messages are not disclosed or suggested. The only "message" mentioned is the "message coming in" which is the datagram input. That message is the one being examined in the IP layer (Fig. 2) to determine whether or not it is encrypted, as discussed earlier because it is an unknown upon arrival. Therefore, that datagram is not a predefined message - it is unknown and, therefore, an undefined message.

There is nothing else in this section that can be reasonably interpreted as an input allegedly representing one of predefined messages. The initiation of the authentication protocol by proxy 50 does not imply predefined messages because that initiation requests a user name and password, and it may include a challenge/response, with respect to an examined datagram source or destination address. Therefore, any such request or challenge is inherently not a predefined message because the message is sent to the instant datagram source or destination address that was just examined. That datagram

address is a destination address for the "message" and that destination address (thus, the message) cannot be defined until the datagram was received. Clearly, every message has at least a destination and content. A message is predefined when at least its destination and content are both predefined. Message content alone is insufficient to define a message. Without a destination to receive the message, content, by itself, has little or no meaning, is not a complete message and is, therefore, not a predefined message. If the Examiner persists in this rejection, Applicants respectfully request that the Examiner point to precisely the language being interpreted as the alleged predefined messages. Consider the other cited section:

> 10. When a dynamic entry is found by a SPI search, the current datagram's SRC address, which is required to ensure that the return datagrams are properly encrypted, will be recorded in the SA only after the AH checking has passed successfully. (This is because if the address is recorded before AH passes then an attacker can cause return packets of an outgoing connection to be transmitted in the clear.) (Minear, col. 11, lines 29-33)

There are no predefined messages based on input disclosed or suggested in the above section. If the Examiner persists in this rejection, Applicants respectfully request that the Examiner point to precisely the language being interpreted as the alleged predefined messages.

Accordingly Minear does not disclose or suggest these limitations of claim 9 and, therefore, does not disclose or suggest claim 9. Mason, cited to teach a call handler, does not cure this deficiency in Minear. Based on MPEP 2143, therefore, a prima facie case of obviousness has not been established against claim 9 on the basis on Minear in view of Mason. Accordingly, the 35 U.S.C. § 103(a) rejection of claim 9 as being un-patentable over Minear and further in view of Mason should be withdrawn and the claim allowed.

Dependent claims 15-21, dependent from claim 9, are also allowable, at least for reasons based on their respective dependencies from allowable base claim 9.

Turning next to independent claim 22 it is allowable for reasons similar to those given above. For example, the Office Action, pg 16, cites Minear, col. 6, lines 13-27 and col. 11, lines 30-33 against the "receiving" step of claim 22 which calls for a "predefined list of function calls." This is the same cite used against the receiving step of claim 5. (see page 16 herein) Just as there were no "predefined messages" of claim 5 disclosed or suggested by these sections, there is no "predefined list of function calls" of claim 21 disclosed or suggested by these sections. Therefore, Minear does not disclose or suggest: "receiving in the at least one processor a first function call from a predefined list of function calls, the predefined list of function calls representing available cryptographic-related functions executable by the at least one processor" as recited in claim 21. If the Examiner persists in this rejection, Applicants respectfully request that the Examiner point to precisely the language being interpreted as the alleged predefined list of function calls.

In addition, without the recited "first function call from a predefined list of function calls", there cannot be a request message based on a non-existent first function call. Therefore, "generating in the at least one processor a request message based on the first function call, the request message representing a request for processing by a cryptographic processing module executed by the at least one processor" is not disclosed or suggested by Minear.

It follows, therefore, that without the request message of the generating step, the transmitting of a non-existent request message cannot take place. Therefore, "transmitting in the at least one processor the request message to the cryptographic processing module" as recited in claim 21 is not disclosed or suggested by Minear. For any or all of the above reasons, the limitations discussed above in Applicants' claim 21 are not disclosed or suggested by Minear.

Mason, cited to teach a call handler, does not cure this deficiency in Minear. Based on MPEP 2143, therefore, a prima facie case of obviousness has not been established against claim 21 on the basis on Minear in view of Mason. Accordingly, the 35 U.S.C. § 103(a) rejection of claim 21 as being un-patentable over Minear and further in view of Mason should be withdrawn and the claim allowed.

CONCLUSION

All rejections in the Office Action have been addressed. In view of the foregoing remarks, reconsideration and allowance of the pending claims are respectfully requested.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 07-2347 and please credit any excess fees to such deposit account.

Respectfully submitted,

By: 
Joel Wall
Reg. No. 25,648

Date: November 15, 2007

Verizon
Patent Management Group
1515 Courthouse Road, Suite 500
Arlington, VA 22201-2909
Tel: 703.351.3586
CUSTOMER NO. 25537